CSE 5243
Homework 2

Soren Goyal, Ohio State University

I. Section 1: Introduction

The main goal of this assignment was to understand the behavior of the kNN classfier using metrics from ROC curves, confusion matrix, etc. The kNN classifier was applied to the two datasets - Iris and Income.

A. kNN on Iris Dataset

The Iris Dataset has four attributes. Each attribute is numeric. It has 150 records, distributed equally among 3 classes - *Setosa, Versicolor and Virginia*. Before the data was used it was preprocessed as follows -

1) Each attribute was centered to 0 and scaled down to a standard deviation of 1.
2) Two datasets were created. One had the original class labels, the other had been binarized for the purpose of generating ROC curves and confusion matrix. The two classes *Versicolor* and *Virginia* were labelled as negatives while *Setosa* was labelled as positive.

Similar tranformation was applied to the Iris test Dataset too.

To predict the class of each of the test records the two functions were used. The first one was `computeDistances(train_data, test_data)` and the second one was `predictLabel(k, distances)`. To classfiy a new records the `predictLabel(k, distances)`

for i in test_data **do**
  for j in train_data **do**
    distance[i,j] ← computeDistance(r,s)
  end fordistances[i,j] ← sort(distance[i,], order = increasing) predictLabel(k, distances[i, 1:k]
end for

The `computeDistances()` function computes the distance of each of the test record from each of the training data record. The distances are sorted and stored for use by `predictLabel(k, distances)`. `predictLabel(k, distances)` takes the number of nearest neighbors (k) as a parameter and predicts the label. The detailed defintions of the functions are given in fig.1 and fig.2.

The reason for separating out the distance computation was the predictions were to be made for different values of k. Distance computation takes $O(mn \log n)$ time, where m is the number of records in test dataset, n is the number of records in training dataset. While prediction takes only $O(mk)$ time, which is considereably less than $O(mn \log n)$.

B. KNN on Income Dataset

The Income dataset has 15 attributes. Based on the analysis during the previous assignment the three of the attributes were dropped and the remaining were split into three groups by the data type they contained - Nominal, Ordinal and Numeric.
The data was preprocessed into two differernt ways. The first way was the same as in the previous assignment, however the results were not satisfactory so they were preprocessed again. The preprocessing was done as follows

- Nominal -

II. Section 2: Evaluation of performance

A. Section II: Analysis of Performance of Classifiers

A number of small variations were tried in each classifer. This section will first define each of those - Variation in Preprocessing

- **Simple Scaling** - The nominal attributes are not touched. Numeric and ordinal attributes are scaled and centered.
- Simple Scaling with Binning - Apart from scaling and centering the numeric and ordinal attributes the nominal attriutes were also modified. Many of the nominal attributes were very skewed. For e.g - In *Native Country* in Income dataset, out of 26 categories, 12 categories had only one record each. In such cases the smalled categories were grouped together. Also certain numeric attributes were also converted to nominal types such as capital gain and capital loss.

Variations in Proximity Metric

- **Euclidean Distance-Simple Dissimilarity** - For numeric attributes euclidean distances were computed, while for nominal attributes simple dissimilarity was computed. These were combined by taking a weighted average.
- **Cosine Similarity-Simple Dissimilarity** - Nominal atributes were treated the same way as the previous type. But for numeric attributes the cosine similarity was computed. Finally, they were combined in the same was as the previous method i.e by taking a weighted average.

Variations in voting the class Let N represent the set of k nearest neighbors. Let $N^(1) \subseteq N$ be set of neighbors belonging to class 1 and $N^(2)$ be the set of neighbors belonging to the other class.

- **Inverse Distance Weighted Voting** - Once the k nearest neighbours have been determined, the probablity of a point belonging to a class is as follows -

$$P(class = i) = \frac{\sum_{r \in N^(i)} \frac{1}{distance(r)}}{\sum_{r \in N} \frac{1}{distance(r)}}$$

- **Positive Class Priority** - In cas eof Income dataset, it was observed that in Fig: ??, the positive class ('¿50k') had a high number of the negative class points around them. So the postive classes were assigned higher weight during computation of probablity.

$$P(class = i) = \frac{\sum_{r \in N^{(i)}} \frac{w_r}{distance(r)}}{\sum_{r \in N} \frac{w_r}{distance(r)}}$$

where, w_r was taken to 3, for postive class and 1 for negative class.

1) Classifier Performance for Iris Dataset

Classifier 1

- Preprocessing - Simple Scaling
- Proximity Metric - Euclidean Distance
- Voting - Inverse Distance Weighted Voting

The confusion matrix for three different values of k is given in Fig: ? ??Analysis??

Fig. 1. Confusion Matrix k = 1 (Datset:Iris Classifier:1)

		Predicted Label		
		Setosa	Versicolor	Virginica
Actual Label	Setosa	20	0	0
	Versicolor	2	20	8
	Virginica	0	4	16

Fig. 2. Confusion Matrix k = 5 (Datset:Iris Classifier:1)

		Predicted Label		
		Setosa	Versicolor	Virginica
Actual Label	Setosa	20	0	0
	Versicolor	1	21	8
	Virginica	0	3	17

Fig. 3. Confusion Matrix k = 10 (Datset:Iris Classifier:1)

		Predicted Label		
		Setosa	Versicolor	Virginica
Actual Label	Setosa	20	0	0
	Versicolor	1	18	11
	Virginica	0	4	16

Classifier 2

- Preprocessing - Simple Scaling (same as before)
- Proximity Metric - Cosine Similarity
- Voting - Inverse Distance Weighted Voting (same as before)

The confusion matrix is given in Fig: ??(use excel) ??Analysis??

Fig. 4. Confusion Matrix k = 1 (Datset:Iris Classifier:2)

		Predicted Label		
		Setosa	Versicolor	Virginica
Actual Label	Setosa	20	0	0
	Versicolor	3	20	7
	Virginica	0	3	16

Fig. 5. Confusion Matrix k = 5 (Datset:Iris Classifier:2)

		Predicted Label		
		Setosa	Versicolor	Virginica
Actual Label	Setosa	20	0	0
	Versicolor	2	24	4
	Virginica	0	3	17

2) Classifier Performance for Income Dataset

To analyze the performance of kNN on the Income dataset two parameters need to be determined. The number of nearest neighbours k and threshold for making the prediction. To optimum k for the classifier would be the k for which the "Area Under the Roc Curve" is maximized. To do this, the roc graphs were computed over the test set for k's ranging from 1 to 100. The Area under the curve increases as k increase. This means that the classifer is getting better at all levels. However after a while it peaks and then plateus. The k at the peak can be chosen as of the best k's.

Classifier 1

- Preprocessing - Simple Scaling
- Proximity Metric - Euclidean Distance
- Voting - Inverse Distance Weighted Voting

The confusion matrix is given in Fig: ??(use excel)
The ROC curve is given in Fig: ??
For this classifier the Area under ROC reaches maximum at k = 38

Classifier 2

- Preprocessing - Simple Scaling
- Proximity Metric - Cosine Similarity
- Voting - Inverse Distance Weighted Voting

The confusion matrix is given in Fig: ??(use excel)
The ROC curve is given in Fig: ??
Max for k = 36??Analysis??

Classifier 3 - Best Performance

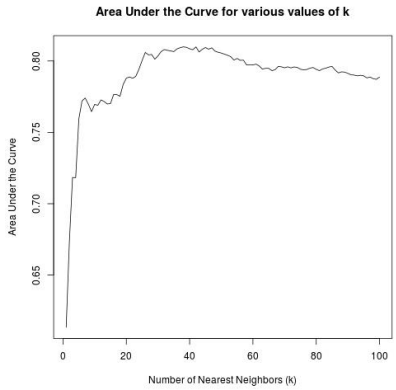
- Preprocessing - Simple Scaling with Binning
- Proximity Metric -
- Voting - Inverse Distance Weighted Voting with Positive Class Priority

The confusion matrix is given in Fig: ??(use excel)
The ROC curve is given in Fig: ??

Fig. 6. Confusion Matrix k = 10 (Datset:Iris Classifier:2)

		Predicted Label		
		Setosa	Versicolor	Virginica
Actual Label	Setosa	20	0	0
	Versicolor	4	20	6
	Virginica	0	4	16

Fig. 7. Plot of Area Under the Roc curve for different values of k for Income Classifier 1. Maxima is achieved at k = 36



??Analysis??

III. Section 3: Comparision with Off-the-Shelf kNN

A. *Off-The-Shelf kNN on Income Dataset*

Fig. 8. ROC curves for different values of k for Income Classifier 1

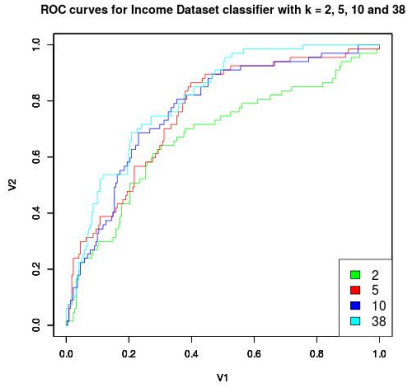


Fig. 9. Plot of Accuracy Vs Threshold Values for k = 38

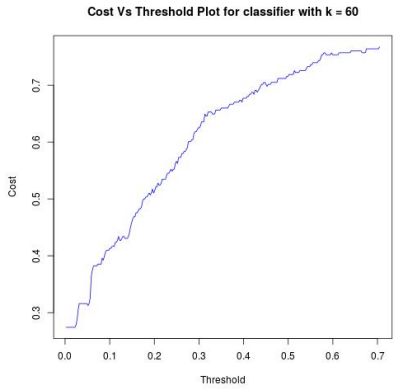


Fig. 10. Plot of Area Under the Roc curve for different values of k for Income Classifier 2. Maxima is achieved at k = 36

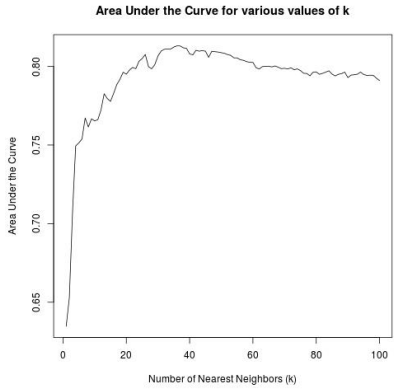


Fig. 11. ROC curves for different values of k for Income Classifier 2

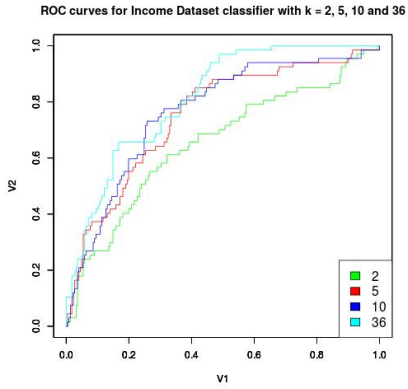


Fig. 12. Plot of Accuracy Vs Threshold Values for k = 36

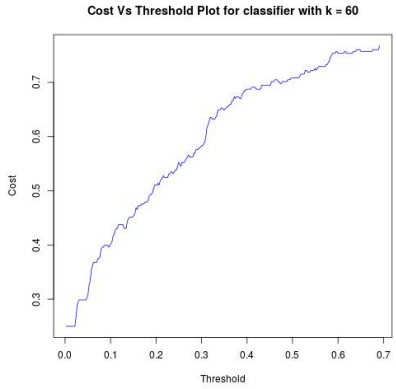


Fig. 13. Plot of Area Under the Roc curve for different values of k for Income Classifier 3. Accuracy keeps getting beyond hundred

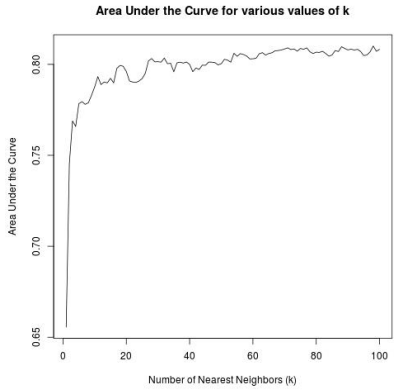


Fig. 14. ROC curves for different values of k for Income Classifier 3

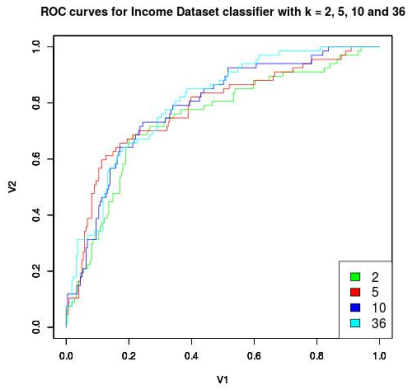


Fig. 15. Plot of Accuracy Vs Threshold Values for k = 40

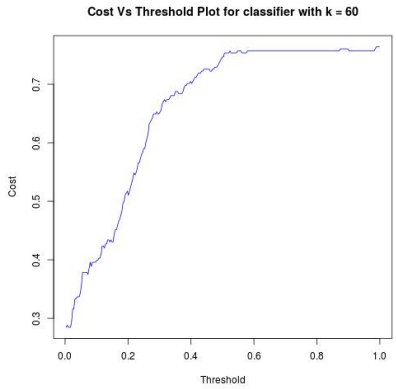


Fig. 16. Plot of Area Under the Roc curve for different values of k for Income Classifier 2. Maxima is achieved at k = 36

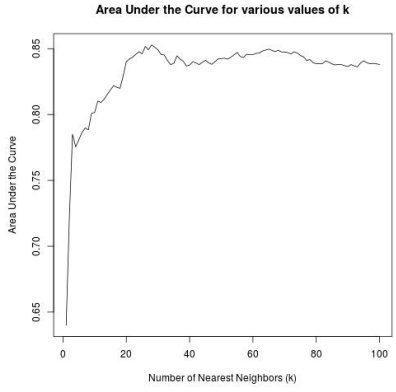


Fig. 17. ROC curves for different values of k for Income Classifier 2

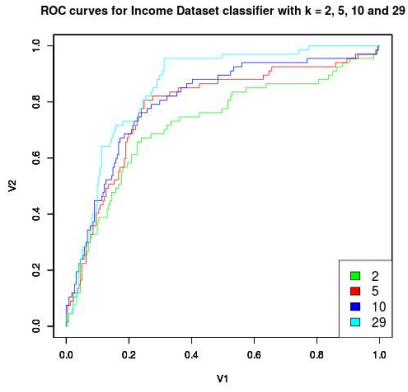


Fig. 18. Plot of Accuracy Vs Threshold Values for k = 36